



## Açık Kaynak Doğal Dil İşleme Kütüphaneleri

Havva Yılmaz<sup>a1,\*</sup>, Semih Yumuşak<sup>a2</sup>,

<sup>a</sup> KTO Karatay Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Konya, Türkiye

Istanbul Sabahattin Zaim Üniversitesi Fen Bilimleri Enstitüsü Dergisi (2021) 3 (1): 81-85

<https://doi.org/10.47769/izufbed.879217>

ORCID <sup>1</sup> 0000-0002-0217-099X; <sup>2</sup> 0000-0002-8878-4991

### YAYIN BİLGİSİ

Yayın geçmişi:

Gönderilen tarih: 14 Şubat 2021

Kabul tarihi: 23 Mart 2021

### Anahtar kelimeler:

Doğal Dil İşleme

Doğal Dil İşleme Kütüphaneleri

Makine Öğrenmesi

### ÖZET

Doğal dil işleme, dil bileşenlerinin hem şekilsel hem de anlamsal olarak analiz edildiği yöntemlere verilen isimdir. Doğal dil işleme yöntemleri sürekli güncellenmekte ve yeni yöntemler geliştirilmektedir. Bu çalışmada, doğal dil işlemede kullanılan güncel ve popüler kütüphaneler ve bu kütüphanelerde kullanılan yöntemler incelenmiştir. Farklı yöntem ve kütüphaneler karşılaştırmalı olarak açıklanmıştır.

## Open Source Natural Language Processing Libraries

### ARTICLE INFO

Article history:

Received: 14 February 2021

Accepted: 23 March 2021

### Key words:

Natural Language Processing

Natural Language Processing

Libraries

Machine Learning

### ABSTRACT

Natural language processing is a collection of methods in which language components are analyzed both syntactically and semantically. This study presents the set of tools and libraries classified as natural language processing methods. Current and popular libraries used in natural language processing and the methods used in these libraries are comparatively explained.

### 1. Giriş

İnsanların kullandığı dilin anlaşılması, analiz edilmesi ve işlenmesi için yapay zeka ve makine öğrenmesi algoritmalarının bir arada kullanıldığı metin ve seslerin dillerin kural yapısına göre işlenmesini sağlayan alandır. Doğal dil işleme iki alt kırılıma bölmek mümkündür. Bu kırılımlar doğal dil anlama ve doğal dil üretme olarak iki temelden oluşmaktadır. Doğal dil anlama metinde geçen bağlamın anlaşılması için kullanılırken, doğal dil üretme yeni bir metin oluşturmak için kullanılır. Birçok doğal dil işleme uygulamasında bu iki alanın beraber kullanıldığı görülmektedir. Doğal dil işleme arama motorları, sanal asistanlar, metin sınıflandırma, metin özetleme, metin çevirisi, duygu analizi, varlık isim tanıma gibi birçok popüler alanı içermektedir (Chowdhary,2020).

### 2. Doğal Dil İşleme Metotları

#### 2.1 Cümlelerin Ögelerine Ayırma (Tokenizasyon)

Metin içindeki cümleleri anlamlı küçük birim parçacıklarına ayırma işlemidir. Cümlelerin ögelerine ayırma işleminin sonucunda çok fazla birim parçacık olduğundan dolayı dizgecik (token) azaltma yöntemleri olarak etkisiz kelimelerin (stop word) atılması, yanlış yazılmış kelimelerin çıkarılması ve kelimelerin kökünü bulma (stemming) ve baş kelimeyi bulma (lemmatization) işlemleri gerçekleştirilebilir (Webster, 1992).

Etkisiz kelimelerin çıkarımı: "de", "da", "ama", "acaba", "sadece", "şu", "ya", "yine", "şey" gibi kelimelerin çıkarılarak dizgecik olarak sayılmaması işlemidir.

Kelimelerin kökünü bulma ve Baş kelimeyi bulma: Kelimelerdeki

\* Sorumlu yazar.

E-mail adresi: [havva.yilmaz@karatay.edu.tr](mailto:havva.yilmaz@karatay.edu.tr) (Havva Yılmaz)

ekleri atarak kökünü bulmaya çalışan yöntemlerdir. Kelimelerin kökünü bulma yöntemi baş kelimeyi bulmaya göre daha basit olup, baş kelimeyi bulma algoritmaları için güçlü bir gramer bilgisi gerekmektedir (Jivani,2011).

## 2.2 Normalizasyon

Metin normalizasyonu, analiz edilen metinde bulunan dağınıklıkları azaltma amacıyla kullanılan yöntemlere verilen isimdir. Normalizasyon yöntemleri arasında; büyük/küçük harf ayarlamaları, gereksiz boşluk veya karakter ayıklama, sayı/yazı düzeltmeleri, kısaltma düzeltmeleri vb. düzenlemeleri içermektedir. İstatistiksel yöntemler ve bazı uzaklık ölçme yöntemlerinin kullanımı bu alanda yaygındır. En bilinen yöntem ise Levenstein mesafe yöntemidir (Sciforce,2020).

## 2.3 Morfoloji

Dizgecik olarak ayrılan cümle bileşenlerini dil bilgisine göre sınıflandırma işlemidir. Örneğin “akılcı” kelimesi için morfolojik analiz “akıl (kök)” –“cı(ek)” şeklindedir. Kelime üzerinde yapılan morfolojik tanımlamalar, yazıda geçen ve farklı ekler alan aynı kelimelerin tespiti için bir gerekliliktir.

## 2.4 Vektörizasyon

Kelime, cümle veya dokümanların vektör olarak ifade edilmesini sağlayan yöntemlere vektörizasyon yöntemleri adı verilir. Makine öğrenmesi algoritmaları veya matematiksel modeller sayılar ile çalıştığından dolayı metinsel ifadeler bu tarz bir dönüşüme ihtiyaç duymaktadır (Sun,2018). Bu amaçla kullanılan farklı vektörizasyon yöntemleri aşağıda açıklanmıştır.

N-Gram: Oluşturulacak olan dizgeciklerin kaç kelimedenden oluşacağını belirler. N dizgeciklerin kaç kelimedenden oluşacağını ifade eder. “Ali okula geldi.” cümlesi için 1-Gram={“Ali” , “okula” , “geldi.”} iken 2-Gram için={“Ali okula” , “okula geldi.”} şeklindedir. Bu metot sayesinde anlamsız olan dizgecikler anlam kazanır. “Türk Hava Yolları” ifadesi için 1-Gram ile oluşan dizgecikler anlamsız iken 3-Gram için oluşan dizgecik anlamlı bir ifade haline gelmiştir (Majumder,2002).

Kelime çantası yaklaşımı (Bag of Words (BOW)): Bir cümledeki kelimelerin dizgecikler arasında geçip geçmediğine göre 0 veya 1 olarak sınıflandırılarak oluşturulan vektörizasyon yöntemidir.

Terim frekansı-ters doküman frekansı (Term Frequency — Inverse Document Frequency (TF-IDF)): Bir kelimenin dokümanda geçme sıklığı ve külliyatta geçme sıklığını göz önünde bulundurarak hesaplanan istatistiksel orandır.

## 2.5 Varlık İsim Tanıma

Metin belgeleri üzerinden kişi, yer, kuruluş, kurum gibi önceden tanımlanmış kategorileri tanımlama işlemidir. 2 temel aşamadan oluşmaktadır. İlk aşamada varlık ismi bulunmakta olup ikinci aşamada ise varlığın kategorisi belirlenmektedir. Varlık kategorileri temelde enamex, numex ve timex olarak ayrılmış olup ilerleyen çalışmalarda bu sayı artırılmış ve kategoriler daha özel hale gelmiştir. Veri etiketleme formatı olarak Raw, IOB, IOB2 gibi yöntemler kullanılmaktadır. Varlık isim tanıma yöntemi müşteri hizmetleri, haber içerik sınıflandırılması, duygu analizi gibi birçok doğal dil işleme uygulamalarında kullanılmaktadır (Liang,2013).

## 2.6 Metin Sınıflandırma

Bir metnin içeriğinin hangi konuya/sınıfa ait olduğunu tahmin eden

doğal dil işleme uygulamaları metin sınıflandırma uygulamaları olarak adlandırılmaktadır. Ayrıca bir metnin yazarının kim olduğuna veya bir mailin içeriğine göre spam mail olup olmadığına karar vermemize yardımcı olan çeşitli uygulamalar da mevcuttur (Aggarwal,2012).

## 2.7 Dil Tanımlama

Dünyada konuşulan 6000 civarında konuşulan dil olduğu düşünülmektedir (David,2020). Bir dilde yazılmış olan dokümanı analiz etmeden önce, o dilin şekilsel yapısını ve sözlüğünü kullanabilmek adına dil tespiti yapmak gerekmektedir. Bu bağlamda kullanılan dil tespit yöntemlerine, dil tanımlama yöntemleri ismi verilmektedir.

## 2.8 Duygu Analizi

Metinlerin olumlu, olumsuz, tarafsız gibi sınıflarından birine aidiyetlik oranı olarak değerlendirilmesi işlemine duygu analizi ismi verilmektedir. Bir konu hakkında insanların düşüncelerinin olumlu veya olumsuz olduğunu değerlendirmek için Twitter, Facebook gibi mikro blog platformlarında birçok uygulaması vardır (Cambria,2017).

## 3. Popüler Doğal Dil İşleme Kütüphaneleri

Bu bölümde popüler doğal dil işleme kütüphaneleri tanımlanmaktadır. Her bir kütüphane başlığının yanında kütüphanenin hangi amaçlarla kullanılabileceğinin aktarıldığı ve sonrasında Tablo 1’ de özetlenen doğal dil işleme yöntemleri listelenmektedir.

*3.1 Natural Language Tool Kit (NLTK) (Varlık isim tanıma, cümlelerin öğelerine ayırma, kelimelerin kökünü bulma, duygu analizi)<sup>2</sup>*

50’nin üzerinde derlem (corpus) ve sözcük kaynağı ile oluşturulmuş, Python programlama dili ile geliştirilmiş, Windows, Mac OS X ve Linux gibi işletim sistemlerinde kullanılabilen açık kaynaklı doğal dil işleme kütüphanesidir. İçerisinde temel metin işleme yöntemlerini (etkisiz kelimelerin atılması, kök bulma, duygu analizi) bulundurmasından dolayı diğer birçok doğal dil işleme kütüphanesi tarafından kullanılmaktadır. (Bird,2019).

*3.2 Spacy (Varlık isim tanıma, cümlelerin öğelerine ayırma, baş kelimeyi bulma)<sup>3</sup>*

61’den fazla dili ve tüm işletim sistemlerini destekleyen, Python programlama dili ile geliştirilmiş, API (Application Programming Interface) dokümanları ve örnek uygulamaları içeren açık kaynaklı doğal dil işleme kütüphanesidir. Dokümanı cümleler haline getirmeden tüm dokümanı işlemek için tasarlanmış bir mimariye sahiptir (Hannibal, Matthew and Montani,2017).

*3.3 AllenNLP (Varlık isim tanıma, cümlelerin öğelerine ayırma, duygu analizi)<sup>4</sup>*

PyTorch kütüphanesi üzerinden yazılmış olup, araştırmacılar için derin öğrenme modellerinin tasarlanması ve hesaplanması amacıyla geliştirilen açık kaynaklı doğal dil işleme kütüphanesidir. Bu kütüphane ile yeni dil anlama modellerinin araştırmacılar tarafından hızlı ve kolay bir şekilde oluşturulabilmesi, web arayüzü veya API desteğiyle geliştirilebilmesi amaçlanmıştır (Gardner, 2017).

*3.4 StanfordNLP (Varlık isim tanıma, cümlelerin öğelerine ayırma, baş kelimeyi bulma, metin sınıflandırma, duygu analizi)<sup>5</sup>*

<sup>2</sup> <http://www.nltk.org>

<sup>3</sup> <https://spacy.io/>

<sup>4</sup> <https://allennlp.org/>

<sup>5</sup> <https://github.com/stanfordnlp/CoreNLP>

Stanford Üniversitesi'ndeki öğrenciler tarafından oluşturulmuş ve Java programlama dili ile geliştirilmiş, 6 dil destekli doğal dil işleme kütüphanesidir. Varlık isim tanıma, duygu analizi gibi konularda basit uygulamaların incelenebileceği bir tanıtım sürümü vardır (McClosky, Bauer,2014).

3.5 Zemberek (Varlık isim tanıma, metin sınıflandırma, dil tanımlama, cümlelerin öğelerine ayırma, kelimelerin kökünü bulma, normalizasyon)<sup>6</sup>

Java programlama dili ile geliştirilmiş açık kaynaklı bir Türkçe doğal dil işleme kütüphanesi olan Zemberek, OpenOffice ve LibreOffice eklentisi olarak kullanılabilir. Yazım denetimi, hatalı kelimeler için öneri, heceleme, hatalı kodlama temizleme gibi işlevlere sahiptir (Zemberek,2007).

3.6 Flair (Metin sınıflandırma, varlık isim tanıma)<sup>7</sup>

PyTorch üzerine yazılmış açık kaynaklı doğal dil işleme çerçevesidir (framework). Günümüzde en gelişmiş kelime vektörü oluşturma metodlarından BERT, GloVe ve ELMo gibi yöntemleri API aracılığıyla kullanıma sunar. Ayrıca arayüzü farklı vektörizasyon metodlarını birlikte kullanabilmeyi sağlar (Akbik, Roland,2018).

3.7 PyTorch (Metin sınıflandırma)<sup>8</sup>

Facebook AI grubu tarafından geliştirilmiş olan bir Python makine öğrenmesi çerçevesidir (framework). Doğal dil işleme ve görüntü işleme alanlarında da birçok uygulaması bulunan açık kaynaklı temel bir kütüphanedir. Birçok ürün için kullanılabilir derecede stabil çalışan aynı zamanda bulut desteği veren zengin bir araç ekosistemi bulunan gelişmiş bir çerçevedir (Paszke,2017).

3.8 TextBlob (Metin sınıflandırma, duygu analizi)<sup>9</sup>

Python programlama dili versiyon 2 ve versiyon 3 için desteği olan temelinde NLTK kütüphanesi kullanılan açık kaynaklı doğal dil işleme kütüphanesidir. Bu kütüphane API desteği sağlamaktadır. Kütüphane eklendikten sonra yazılan metnin olumlu-olumsuz durumuna göre kullanıcıya 0-1 aralığında bir değer döndürmektedir (Loria,2013).

3.9 Gensim (Vektörizasyon)<sup>10</sup>

Python programlama dilinde Numpy ve Scipy kütüphaneleri üzerine kurulu birçok vektörizasyon yöntemini içeren popüler doğal dil işleme algoritmalarının bir uygulamasını sunan bir kütüphanedir. Word2vec gibi popüler doğal dil işleme algoritmalarının bir uygulamasını sunar (Radim, Sojka,2008).

3.10 Scikit-Learn (Vektörizasyon, metin sınıflandırma, kelimelerin kökünü bulma)<sup>11</sup>

Python programlama dili ile yazılmış regresyon, boyutsallığın azaltılması ve kümelemeyi içeren makine öğrenimi görevleri gibi birçok temel algoritmayı içeren gelişmiş bir makine öğrenmesi kütüphanesidir. Python kütüphaneleri olan NumPy ve Matplotlib kütüphaneleri üzerine inşa edilmiştir (Buitinck, Louppe,2013).

3.11 Turkish NLP, Turkish Stemmer, Turkish Lemmatizer, Zemberek Parser (Cümlelerin öğelerine ayırma, kelimelerin kökünü bulma, baş

kelimeyi bulma)<sup>12,13,14,15</sup>

Cümlelerin Öğelerine Ayırma, kelimelerin kökünü bulma ve baş kelimeyi bulma işlemleri için geliştirilmiş açık kaynaklı Türkçe doğal dil işleme kütüphaneleridir. Bu kütüphaneler ile metinlerin boşluklarını düzenleme, yazım hatalarını düzenleme, kelimenin kökeninin Türkçe olup olmadığını algılama, heceleme işlemleri gerçekleştirilebilir (Çetinkaya,2018), (Onaldi,2018), (Koksal,2018), (Bora,2020).

3.12 Pyspellchecker (Normalizasyon)<sup>16</sup>

Yanlış yazılmış olan kelimelerin düzeltilmesi için geliştirilmiş açık kaynaklı bir Python kütüphanesidir. Hamming, Manhattan, Levenstein gibi mesafe ölçüm yöntemleri bulunmaktadır. Pyspellchecker kütüphanesi iki kelimenin arasındaki mesafeyi ölçmek için varsayılan olarak Levenstein Mesafe metodunu kullanmaktadır (Barrus,2018).

3.13 ITU Türkçe Doğal Dil İşleme Yazılım Zinciri (Varlık isim tanıma, cümlelerin öğelerine ayırma, normalizasyon, kelimelerin kökünü bulma)<sup>17</sup>

Tüm doğal dil işleme sürecini yönetebileceğimiz açık kaynaklı olmayan fakat API desteği ile doğal dil işleme projelerine destek veren bir çalışmadır. Bu çalışma Türkçe karakter dönüştürücü (deasciifier), cümlelerin öğelerine ayırma, yazım denetleyici (spell checker), varlık isim tanıma gibi araçlardan oluşan bir platformdur. Bu platform hem bir web arayüzüne hem de bir uygulama arayüzüne sahip olduğu için farklı seviyedeki araştırmacılar bu platformdan faydalanabilir. Doğal dil işleme yöntemlerini uygulayabileceğimiz bir ön gösterimi bulunmaktadır (Eryiğit,2014).

3.14 Fast.ai (Normalizasyon, cümlelerin öğelerine ayırma)<sup>18</sup>

Hem makine öğrenmesi hem de derin öğrenme algoritmalarının bir uygulama kütüphanesidir. Jeremy Howard tarafından San Francisco Üniversitesi Veri Enstitüsü'nde çevrimiçi kurs olarak da sunulan bir derin öğrenme platformudur. Özyinelemeli ağlar gibi doğal dil işleme alanında yaygın derin öğrenme modellerini oluşturabileceğimiz bir kütüphanedir (Howard,2017).

3.15 SentiTurkNet (Duygu analizi)<sup>19</sup>

Özellikle duygu analizi yöntemi için geliştirilen açık kaynaklı bir Türkçe doğal dil işleme kütüphanesidir. Dehkharghani tarafından geliştirilmiş olan bu kütüphanede Türkçe kelimelerin duygu kutuplarını belirten bir sözlük oluşturulmuştur. Bu sözlükte metnin sınıflandırılması öfke, nefret, korku, mutluluk, üzüntü, sürpriz olarak adlandırılan altı duyguya göre yapılmaktadır (Dehkharghani,2015).

Doğal Dil İşleme Yöntemleri	Kullanılan Kütüphaneler
Varlık İsim Tanıma	NLTK,Spacy, AllenNLP, StanfordNLP, Zemberek
Metin Sınıflandırma	Zemberek, Flair, PyTorch
Dil Tanımlama	TextBlob, Zemberek

<sup>6</sup> <https://github.com/ahmetaa/zemberek-nlp>

<sup>7</sup> <https://github.com/flairNLP/flair>

<sup>8</sup> <https://github.com/pytorch/pytorch>

<sup>9</sup> <https://textblob.readthedocs.io/en/dev/>

<sup>10</sup> <https://pypi.org/project/gensim/>

<sup>11</sup> <https://scikit-learn.org/stable/>

<sup>12</sup> <https://pypi.org/project/turkishnlp/>

<sup>13</sup> <https://github.com/otuncelli/turkish-stemmer-python/>

<sup>14</sup> <https://github.com/akoksal/Turkish-Lemmatizer>

<sup>15</sup> [https://github.com/kemalcanbora/zemberek\\_parser](https://github.com/kemalcanbora/zemberek_parser)

<sup>16</sup> <https://pypi.org/project/pyspellchecker/>

<sup>17</sup> <http://tools.nlp.itu.edu.tr/>

<sup>18</sup> <https://www.fast.ai/>

<sup>19</sup> <http://myweb.sabanciuniv.edu/rdehkharghani/sentiturknet-3/>

Vektörizasyon	Gensim, Scikit-Learn
Cümlelerin öğelerine ayırma, Kelimelerin kökünü bulma, Baş kelimeyi bulma	Zemberek,Turkish NLP, Turkish Stemmer,Turkish Lemmatizer, Zemberek Parser
Morfoloji	Zemberek,ITU Turkish Natural Language Processing Pipeline(API olarak açık)
Normalizasyon	Zemberek, Fast.ai, Pyspellchecker
Duygu Analizi	SentiTurkNet, TextBlob

**Tablo 1.** Doğal dil işleme yöntemleri ve kullanılan kütüphaneler

Tablo I' de doğal dil işleme yöntemleri ve bu yöntemlerde kullanılan popüler algoritmalara yer verilmiştir. Doğal dil işleme alanında çalışma yapacak araştırmacılar bu tabloyu kullanarak çalışacakları alanlarda kullanabilecekleri kütüphanelere kolayca erişebilirler.

#### 4. Sonuç ve Tartışma

Bu çalışmada yapay zeka ve dilbilim alt kategorisi olan doğal dil işleme alanına ait yöntemler ve bu alanda kullanılan popüler kütüphaneler incelenmiştir. Bu alanda yapılacak çalışmalar için ilk adım olan kütüphane seçiminin doğru ve hızlı bir şekilde yapılması amacıyla yürütülecek proje alt dalına uygun kütüphanelerin bulunabilmesi özet listeler sunulmaktadır. Doğal dil işlemenin evrensel kurallara göre yapıldığı NLTK, Spacy ve StanfordNLP gibi kütüphaneler birçok diğer doğal dil işleme kütüphanelerine temel oluşturmaktadır. Ayrıca, basit makine öğrenmesi tekniklerinin kullanıldığı Scikit-Learn ve Gensim gibi kütüphaneler, yazı verisini analiz etmek ve bilgi çıkarımı yapmak amaçlı temel gereklilikleri sağlamaktadır. Bununla beraber yine makine öğrenmesi ve derin öğrenme yöntemleri ile analizler gerçekleştirilmesini sağlayan AllenNLP ve benzeri kütüphaneler yüksek doğrulukta varlık tanıma, ilişki tespiti gibi yöntemleri kullanmak üzere tasarlanmıştır. Bu kütüphanelerin yanı sıra hatalı kelime düzeltme, yazı denetimi, cümlelerin öğelerine ayırma, kelimelerin kökünü bulma ve baş kelimeyi bulma gibi yöntemlerin uygulanabileceği Turkish NLP, Zemberek, Turkish Stemmer, Turkish Lemmatizer ve ITU Turkish NLP Pipeline gibi açık kaynak Türkçe kütüphaneler bulunmaktadır.

#### Kaynaklar

- Aggarwal, C. C., & Zhai, C. (2012). A survey of text classification algorithms. In *Mining text data* (pp. 163-222). Springer, Boston, MA.
- Akbik, Roland, (2018). Flair A very simple framework for state-of-the-art NLP, 2020. [Çevrimiçi]: <https://github.com/flairNLP/flair>
- Barrus, (2018). Pyspellchecker Pure python spell checker based on work by Peter Norvig, 2020. [Çevrimiçi]: <https://pypi.org/project/pyspellchecker/>
- Bora, (2020). Zemberek Parser , 2019. [Çevrimiçi]: [https://github.com/kemalcambora/zemberek\\_parser](https://github.com/kemalcambora/zemberek_parser)
- Buitinck, Louppe, (2013). Scikit-learn 0.23.2 Machine Learning in Python, 2020. [Çevrimiçi]: <https://scikit-learn.org/stable/>
- Bird, (2019). Natural Language Toolkit, 2020. [Çevrimiçi]: <http://www.nltk.org>
- Cambria, E., Poria, S., Gelbukh, A., & Thelwall, M. (2017). Sentiment analysis is a big suitcase. *IEEE Intelligent Systems*, 32(6), 74-80.

- Chowdhary, (2020). Natural language processing. In *Fundamentals of Artificial Intelligence* (pp. 603-649). Springer, New Delhi.
- Çetinkaya, (2018). Turkish NLP, 2020. [Çevrimiçi]: <https://pypi.org/project/turkishnlp/>
- David, 2020. How many languages in the world. [Çevrimiçi]: <https://www.ethnologue.com/guides/how-many-languages>
- Dehkharghani, Rahim & Saygin, Yucel & Yanikoglu, Berrin & Oflazer, Kemal. (2015). SentiTurkNet: a Turkish polarity lexicon for sentiment analysis. *Language Resources and Evaluation*. 50. 10.1007/s10579-015-9307-6.
- Eryiğit, G. (2014, April). ITU Turkish NLP web service. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 1-4).
- Gardner, (2017). A natural language processing platform for building state-of-the-art models. 2020. [Çevrimiçi]: <https://allennlp.org/>
- Honnibal, Matthew, (2017). Spacy Industrial-Strength Natural Language Processing in Python, 2020. [Çevrimiçi]: <https://spacy.io/>
- Howard, (2017). Fast.ai , 2020. [Çevrimiçi]: <https://www.fast.ai/>
- Jivani, A. G. (2011). A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, 2(6), 1930-1938.
- Koksal, (2018). GitHub - akoksal/Turkish-Lemmatizer: Lemmatization for Turkish Language, 2018. [Çevrimiçi]: <https://github.com/akoksal/Turkish-Lemmatizer>
- Liang, J., Koperski, K., Dhillon, N. S., Tusk, C., & Bhatti, S. (2013). U.S. Patent No. 8,594,996. Washington, DC: U.S. Patent and Trademark Office.
- Lorai, (2013). TextBlob: Simplified Text Processing, 2020. [Çevrimiçi]: <https://textblob.readthedocs.io/en/dev/>
- Majumder, P., Mitra, M., & Chaudhuri, B. B. (2002). N-gram: a language independent approach to IR and NLP. In *International conference on universal knowledge and language*.
- McClosky, Bauer, (2014). Stanford CoreNLP: A Java suite of core NLP tools. 2020. [Çevrimiçi]: <https://github.com/stanfordnlp/CoreNLP>
- Onaldi, (2018). Turkish Stemmer, 2019. [Çevrimiçi]: <https://github.com/otuncelli/turkish-stemmer-python/>
- Paszke, (2017). Pytorch/pytorch: Tensors and Dynamic neural networks in Python with strong GPU acceleration, 2020. [Çevrimiçi]: <https://github.com/pytorch/pytorch>
- Radim, Sojka, (2008). Gensim 3.8.3 Python framework for fast Vector Space Modelling, 2020. [Çevrimiçi]: <https://pypi.org/project/gensim/>
- Sciforce, (2020). Text Preprocessing for NLP and Machine Learning Tasks [Çevrimiçi]: <https://medium.com/sciforce/text-preprocessing-for-nlp-and-machine-learning-tasks-3e077aa4946e>
- Sun, Q., Wang, B., Gu, Z., & Fu, Y. (2018). VECTORIZATION METHODS IN RECOMMENDER SYSTEM.
- Webster, (1992). Tokenization as the initial phase in NLP. In *COLING 1992 Volume 4: The 15th International*

Conference on Computational Linguistics.

<https://github.com/ahmetaa/zemberek-nlp>

Zemberek, (2007). NLP tools for Turkish. [Çevrimiçi]: